REMARKS

In the final Office Action, the Examiner rejected a handful of the pending claims on the basis of 35 USC § 251 asserting that they recapture material that was given up during prosecution of the parent application. Some of the claims were also rejected under 35 USC §102(b) as being anticipated by White. For clarity, these rejections will be discussed separately in the remarks below.

REJECTION OF CLAIMS 40-53 AND 60-68 UNDER 35 USC §251

In the final Office Action, the Examiner rejected claims 40-53 and 60-68 under 35 USC §251 as improperly recapturing subject matter cancelled in the application for the patent upon which the present reissue is based. To simplify the issues in this case, the language of dependent claim 44 has been added to each of the independent that were rejected on the basis of recapture (i.e., claims 40, 50, 52, 60, 64 and 67). For the reasons set forth below, it is respectfully submitted that this language should address the Examiner's recapture concerns. It is noted, however, that the Applicant believes that the claims as previously pending did not constitute recapture under 35 USC 251 and that the Applicant expects to pursue claims of that scope in a continuation application.

The various claims rejected on the basis of recapture all relate to methods of or devices for *decrypting* data packets. It is the undersigned's understanding that the recapture rejection is based on the position that during prosecution of claims relating to *encryption* of data packets, amendments were made which required the generation of a "new" address header. This has apparently been interpreted as requiring that the resulting claims then required the presence of both the "new" address header and the "old" address header. Therefore, it is understood that the Examiner's position is that the packets to be decrypted must include two different headers (i.e., the header of the encrypted packet and the header of the original packet).

Initially, it is respectfully submitted that as a matter of law, amendments to claims directed at "encrypting" data packets should not in any way influence the scope of broadening amendments directed at methods of "decrypting" data packets since these are very different

processes. Additionally, regardless of the Examiner's position on that legal issue, it is respectfully submitted that previously pending claim 44 (as well as previously pending claims 45-47 and 63) positively recited the two header structure that is believed to be sought by the Examiner. That language has now been incorporated into each of the independent claims 40, 50, 52, 60, 64 and 67. Accordingly it is respectfully submitted that the rejection of claims 40-53 and 60-68 under 35 USC §251 should be withdrawn.

It is noted that the claim language of claim 44 has been edited somewhat to improve its readability in the context of the independent claims. To assist the Examiner in reviewing the current claim language, independent claim 40 is reproduced below with the amendments thereto being underlined. Similar language has also been added to claims 50, 52, 60, 64 and 67.

> 40. A method of decrypting data packets, comprising:
> receiving a data packet from a source for a destination, the data packet including a header section and a data section, the header section storing a source identifier identifying a broadcast address of the source and a destination identifier identifying a broadcast address of the destination;
> determining whether the data packet is encrypted upon reference to at least one of the source and destination identifiers; and
> if the data packet is encrypted, decrypting the data packet to produce a decrypted data packet, <u>wherein the data section of the data packet includes an encrypted header section and an encrypted data section, and after decryption, the decrypted encrypted header section is used as the header for the decrypted data packet.</u>

<u>REJECTION OF CLAIMS 40-41, AND 44-53 UNDER 35 USC §102</u>

40-41, and 44-53 stand rejected under 35 USC §102(b) as being anticipated by White. These rejections are respectfully traversed. Independent claims 40, 50 and 52 each require that the received packet includes the "broadcast" addresses of both the source and the destination. By way of example, in claim 40, specifically requires:

> a header section storing a source identifier identifying a <u>broadcast</u> address of the source and a destination identifier identifying a <u>broadcast</u> address of the destination; *(emphasis added)*

It is the undersigned understanding that both the Applicant and the Examiner agree that the prior art White reference teaches that the unencrypted header must identify the **nodes** that acts as gateway to the WANs. (See, column 4, lines 10-14 of White.) In contrast, the claims require that the header must include the "broadcast" addresses of the source and destination. It is

respectfully submitted that a node address is well understood in the art to be very different than a "broadcast" address.

The difference between the Examiner's position and the Applicant's position appears to relate to the definition of the term "broadcast" address. It is respectfully submitted that the term "broadcast address", as well as it's meaning are extremely well known to those in the art. Specifically, the broadcast address represents a particular network, but not any particular host. See, e.g., Col. 6, lines 54-57. In contrast, a node address is understood to represent a particular host or node on the network. The Site Address of the node utilized in the header taught by White identifies just such a host. See, e.g., Col. 4, lines 10-15 of White.

By way of background, it is noted that in the context of an IP address, a broadcast address is simply an IP address that contains all 1s or 0s in the host portion of the IP address. See, for example, the definition set forth in the accompanying sheet. It is further noted that this is exactly the same context as the term "broadcast address" was used in the specification (see, e.g., col. 6, lines 54-57). This is also the same context that the term was used in the "Understanding IP Addressing" reference cited by the Examiner.

Independent claims 40, 50 and 52 each further require:

> determining whether the data packet is encrypted upon reference to at least one of the source and destination identifiers [*which are defined above as broadcast addresses*]; and
> if the data packet is encrypted, decrypting the data packet to produce a decrypted data packet, *(commentary added)*

That is, the claims require that the determination of whether the data packet is to be decrypted is based at least in part on reference to at least one of the *broadcast addresses*. It is respectfully submitted that nothing in the White reference remotely suggests that the determination of whether to decrypt a packet is based upon a reference to a broadcast address that serves as a source or destination identifier in the packet header.

In view of the foregoing, it is respectfully submitted that nothing whatsoever in White either discloses or reasonably suggests the combinations required by any of claims 40-41, and 45-53 and that those claims are patentable over the art of record for at least that reason.

Additionally the dependent claims 41, 45-49, 51 and 53 each require other elements that when taken in the context of the claimed invention further patentably distinguish the art of record.

REJECTION OF CLAIMS 32 AND 33 UNDER 35 USC §102

Claim 32 pertains to a method of encryption that produces a modified data packet including a header portion that stores an identifier of the network of the source of the data packet. Similarly, claim 33 pertains to a method of encryption that produces a modified data packet including a header portion that stores an identifier of the network of the destination of the data packet. Thus, both claims 32 and 33 each specifically require that an identifier of a network be provided in a header portion of the data packet.

A single host cannot be identified through the mere identification of a network. As described above, White requires that the header identify an actual node via which the packet enters and leaves the network, as described above. In contrast, the invention of claims 32 and 33 prevents using the address of a particular node, particularly a node responsible for encryption or decryption of the data packet. The invention of claims 32 and 33 therefore provides greater protection against tapping into the network to decipher the nature of the information transmitted. Accordingly, Applicant respectfully submits that White's disclosure of a header that identifies a node via which a packet enters and leaves a network does not anticipate claims requiring that a header section of a data packet include an identifier of a network of the destination or source of the data packet. Therefore, it is respectively submitted that claims 32 and 33 are not anticipated by White, and that the pending rejection of these claims should be reversed for the reasons set forth.

SUMMARY

In view of the forgoing, it is respectfully submitted that this case is now in condition for allowance. If there are any issues remaining after consideration of this response, the Examiner is respectfully requested to contact the undersigned attorney at the telephone number listed below.

In view of the timing of the filing of this amendment, a notice of appeal was previously filed to maintain the pendency of this case. Accordingly it is believed that no extensions of time are required. However, Applicants hereby provisionallypetition for any extension(s) of time which may be required to maintain the pendency of this case, and any required fee for such

extension or any further fee required in connection with the filing of this Amendment may be charged to Deposit Account No. 500388 (Order No. SUN1P342R).

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

Steve D Beyer
Reg. No. 31,234

BEYER WEAVER & THOMAS, LLP
P.O. Box 778
Berkeley, CA 94704-0778
Tel: (650) 961-8300

APPENDIX

1. (Once Amended) A method for transmitting and receiving packets of data via **[a]** **an** internetwork for a first host computer on a first computer network to a second host computer on a second computer network, the first and second computer networks including, respectively, first and second bridge computers, each of said first and second host computers and first and second bridge computers including a processor and a memory for storing instructions for execution by the processor, each of said first and second bridge computers further including memory for storing at least one predetermined encryption/decryption mechanism and information identifying a predetermined plurality of host computers as hosts requiring security for packets transmitted between them, the method being **carried** **[carded]** out **[be]** **by** means of the instructions stored on said respective memories and including the steps of:

(1) generating, by the first host computer, a first data packet for transmission to the second host computer, a portion of the **first** data packet including information representing an internetwork address of the first host computer and internetwork address of the second host computer;

(2) in the first bridge computer, intercepting the first data packet and determining whether the first and second host computers are among the predetermined plurality of host computers for which security is required, and if not, proceeding to step 5, and if so, proceeding to step 3;

(3) encrypting the first data packet in the first bridge computer;

(4) in the first bridge computer, generating and appending to the **encrypted** first data packet an encapsulation header, including:

(a) key management information **[identifying]** **providing a mechanism for identifying** the predetermined encryption method, and

(b) a new address header representing the source and destination for the **first** data packet, hereby generating a modified **first** data packet;

(5) transmitting the **first** data packet **or the modified first data packet** from the first bridge computer via the internetwork to the second computer network;

(6) intercepting the **first** data packet **or the modified first data packet** at the second bridge computer;

(7) in the second bridge computer, **if the encapsulation header has been appended to the first data packet,** reading the encapsulation header, and determining therefrom whether the **first** data packet was encrypted, **[and if not, proceeding to**

step 10, and if so, proceeding to step 8] and if it is determined that the first data packet has been encrypted, proceeding to step 8 and otherwise proceeding to step 10;

(8) in the second bridge computer, determining which encryption mechanism was used to encrypt the first data packet;

(9) decrypting the first data packet by the second bridge computer;

(10) transmitting the first data packet from the second bridge computer to the second host computer[,] ; and

(11) receiving the unencrypted **first** data packet at the second host computer.

2. (Once Amended) The method of claim 1, wherein the new address header for the modified **first** data packet includes the address of the second bridge computer.

3. (Once Amended) The method of claim 2, wherein the new address header for the modified **first** data packet includes an identifier of the second bridge computer.

4. (Once Amended) The method of claim 1, wherein the new address header of the modified **first** data packet includes the address of the second host computer.

5. (Once Amended) The method of claim 4, wherein the new address header for the modified **first** data packet includes an identifier of the second bridge computer.

6. (Once Amended) A system for automatically encrypting and decrypting data packets transmitted from a first host computer on a first computer network to a second host computer on a second computer network, including:

a first bridge computer coupled to the first computer network for intercepting data packets transmitted from said first computer network, the first bridge computer including a first processor and a first memory storing instructions for executing encryption of data packets according to a predetermined encryption/decryption mechanism;

a second bridge computer coupled to the second computer network for intercepting data packets transmitted to said second computer network, the second bridge computer including a second processor and a second memory storing instructions for executing decryption of the data packets;

said first host computer including a third processor and a third memory including instructions for transmitting a first **[said]** data packet from said first host to said second host;

a **first** table stored in said first memory including a correlation of at least one of the first host computer and the first network with one of the second host computer and the second network, respectively;

instructions stored in said first memory for intercepting said first data packet before departure from said first network, determining whether said correlation is present in said **first** table, and if so, then executing encryption of said first data packet according to said predetermined encryption/decryption mechanism, generating a new address header **including a mechanism for identifying said predetermined encryption/decryption mechanism** and appending said new address header to said **encrypted** first data packet, thereby generating a modified first data packet, and transmitting said modified first data packet on to the second host computer;

a second table stored in said second memory including a correlation of at least one of the first host computer and the first network with one of the second host computer and the second network, respectively; and

instructions stored in said second memory for intercepting said **modified** first data packet upon arrival at said second network, determining whether said correlation is present in said **second** table, and if so, then executing decryption of said first data packet according to said predetermined encryption/decryption mechanism, and transmitting the first data packet to the second host computer.

7. (Once Amended) **[The method of claim 6,] A system for automatically encrypting and decrypting data packets transmitted from a first host computer on a first computer network to a second host computer on a second computer network, including:**

a first bridge computer coupled to the first computer network for intercepting data packets transmitted from said first computer network, the first bridge computer including a first processor and a first memory storing instructions for executing encryption of data packets according to a predetermined encryption/decryption mechanism;

a second bridge computer coupled to the second computer network for intercepting data packets transmitted to said second computer network, the second

bridge computer including a second processor and a second memory storing instructions for executing decryption of the data packets;

said first host computer including a third processor and a third memory including instructions for transmitting a first data packet from said first host to said second host;

a first table stored in said first memory including a correlation of at least one of the first host computer and the first network with one of the second host computer and the second network, respectively;

instructions stored in said first memory for intercepting said first data packet before departure from said first network, determining whether said correlation is present in said first table, and if so, then executing encryption of said first data packet according to said predetermined encryption/decryption mechanism, generating a new address header and appending said new address header to said encrypted first data packet, thereby generating a modified first data packet, and transmitting said modified first data packet on to the second host computer, wherein said new address header includes **[the]** internetwork broadcast addresses of the first and second computer networks**[.];**

a second table stored in said second memory including a correlation of at least one of the first host computer and the first network with one of the second host computer and the second network, respectively; and

instructions stored in said second memory for intercepting said modified first data packet upon arrival at said second network, determining whether said correlation is present in said second table, and if so, then executing decryption of said first data packet according to said predetermined encryption/decryption mechanism, and transmitting the first data packet to the second host computer.

8. The method of claim 7, wherein said new address header includes an identifier of the second bridge computer.

9. The method of claim 6, wherein said new address header includes the address of the second host computer.

10. The method of claim 9, wherein said new address header includes an identifier of the second bridge computer.

11. (Once Amended) A method for transmitting and receiving packets of data via an internetwork from a first host computer on a first computer network to a second host computer on a second computer network, **[the first and second computer networks,]** each of said first and second host computer networks, each of said first and second host computers including a processor and a memory for storing instructions for execution by the processor, each said memory storing at least **[on]** **a** predetermined encryption/decryption mechanism and a source/destination table identifying a predetermined plurality of sources and destinations requiring security for packets transmitted between them, the method being **carried** **[carded]** out by means of the instructions stored in said respective memories and including the steps of:

(1) generating, by the first host computer, a first data packet for transmission to the second host computer, a portion of the **first** data packet including information representing an internetwork address of a source of the **first data** packet and an internetwork address of a destination of the **first data** packet;

(2) in the first host computer, determining whether the source and destination of the first data packet are among the predetermined plurality of sources and destinations identified in said source/destination table for which security is required, and if not, proceeding to step 5, and if so, proceeding to step 3;

(3) encrypting the first data packet in the first host computer;

(4) in the first host computer, generating and appending to the **encrypted** first data packet an encapsulation header, including:

(a) key management information **providing a mechanism for** identifying the predetermined encryption method, and

(b) a new address header identifying the source and destination for the **first** data packet**, hereby generating a modified first data packet**;

(5) transmitting the **first** data packet **or the modified first data packet** from the first host computer via the internetwork to the second computer network;

(6) in the second host computer, **if the encapsulation header has been appended to the first data packet,** reading the encapsulation header, and determining therefrom whether the first data packet was encrypted, and if **the first data packet was not encrypted [not]**, ending the method, and if **[so]the first data packet was encrypted**, proceeding to step 7;

(7) in the second host computer, determining which encryption mechanism was used to encrypt the first data packet; and

(8) decrypting the first data packet by the second host computer.

12. (Once Amended) The method of claim 11, wherein the new address header for the modified **first** data packet includes internetwork broadcast addresses of the first and second computer networks.

13. The method of claim 11, wherein the source/destination table includes data identifying internetwork addresses of the first and second host computers.

14. (Once Amended) A system for automatically encrypting and decrypting data packets transmitted from a first host computer on a first computer network **[and having a first host computer on a first computer network and] , the first host computer** having **a** first processor and a first memory, via an internetwork to a second host computer on a second computer network **[and having a second host computer on a second computer network and] , the second host computer** having a second processor and a second memory, the system including:

security data stored **in** said first and **second** memories indicating that data packets meeting at least one predetermined criterion are to be encrypted;

a predetermined encryption/decryption mechanism stored in said first and second memories;

a decryption key stored in said second memory;

instructions stored in said first memory for determining whether to encrypt **one or more** data packets, by determining whether said **at least one** predetermined criterion is met by said **one or more data packets [data packet];**

instructions stored in said first memory for executing encryption according to said predetermined encryption/decryption mechanism of at least a first **[said data packet] one of said one or more data packets**, when said **at least one predetermined** criterion is met, for generating a new address header for said first data packet and for appending an encapsulation header to said first data packet and transmitting said first data packet to said second host, **said new address header identifying broadcast addresses of the first and second computer networks,** said encapsulation header including at least said new address header; **and**

instructions stored in said second memory for receiving said first data packet, determining whether it has been encrypted by reference to said security data in said second memory, and if so then determining which encryption/decryption mechanism was used for encryption, and decrypting said **first** data packet by use of said decryption key.

15. (Once Amended) The system of claim 14, wherein:

said security data comprises correlation data stored in each of said first and second memories **[identifying at least one of said first and second memories]** identifying at least one of said first host computer and said first network correlated with at least one of said second host computer and said second network;

the system further including instructions stored in said first memory for determining whether to encrypt data packets by inspecting for a match between source and destination addresses of said data packets with said correlation data.

16. (Once Amended) A system for automatically encrypting data packets for transmission from a first host computer on a first computer network to a second host computer on a second computer network, said first host computer including a first processor and a first memory including instructions for transmitting said data packets from said first host to said second host, the system including:

a bridge computer coupled to the first computer network for intercepting at least a first **[said]** data packet transmitted from said first computer network, said bridge computer including a second processor and a second memory storing instructions for executing encryption of said first data packet according to a predetermined encryption/decryption mechanism;

information stored in said second memory correlating at least one of the first host computer and the first network with one of the second host computer and the second network, respectively; **and**

instructions stored in said second memory for intercepting said first data packet before departure from said first network, determining whether said correlation is present, and if so, then executing encryption of said first data packet according to said predetermined encryption/decryption mechanism, generating a new address header **including a mechanism for identifying said predetermined encryption/decryption mechanism** and appending said new address header to said first data packet, thereby generating a modified first data packet on to the second host computer.

17. (Once Amended) A method for transmitting packets of data via an internetwork from a first host computer on a first computer network to a second host computer on a second computer network, the first computer networks including a first bridge computer, each of said first and second host computers and said bridge computer further including memory storing at least one predetermined encryption/decryption mechanism and information identifying a predetermined plurality of host computers as hosts requiring security for packets transmitted between them, the method being carried out according to the instructions stored in said respective memories and including the steps of:

(1) generating, by the first host computer, a first data packet for transmission to the second host computer, a portion of the **first** data packet including information representing an internetwork address of the first host computer and an internetwork address of the second host computer.

(2) in the first bridge computer, intercepting the first data packet and determining whether the first and second host computers are among the predetermined plurality of host computers for which security is required, and if not, proceeding to step 5, and if so, proceeding to step 3;

(3) encrypting the first data packet in the first bridge computer;

(4) in the first bridge computer, generating and appending to the first data packet an encapsulation header, including:

(a) key management information **providing a mechanism for** identifying the predetermined encryption method, and

(b) a new address header representing the source and destination for the data packet, thereby generating a modified **first** data packet; and

(5) transmitting the **first** data packet **or the modified first data packet** from the first bridge computer via the internetwork to the second computer network.

18. (Once Amended) **A system for automatically decrypting data packets transmitted from a first computer to a second computer, the system comprising:**

a bridge coupled to the second computer for intercepting a data packet from the first computer, the data packet having an address header and a body, the address header including broadcast addresses of the first and second computers, the bridge including a processor and a memory that stores instructions for decrypting data packets;

information stored in the memory of the bridge correlating the first and second computers; and

instructions stored in the memory for intercepting the data packet, determining whether the information stored in the memory of the bridge correlates the first and second computers, and if so, decrypting at least a portion of the data packet to generate a new data packet including a new address header, and transmitting the new data packet onto the second computer.

19. (Once Amended) The system of claim 18, wherein the data packet includes the new data packet in encrypted form.

20. (Twice Amended) A system for automatically decrypting data packets transmitted from a first computer to a second computer, the system comprising:

a bridge coupled to the second computer for intercepting a data packet from the first computer, the data packet including a header storing key management information providing a mechanism for identifying an encryption method used to encrypt the data packet, the bridge including a processor and a memory that stores instructions for decrypting data packets;

information stored in the memory of the bridge correlating the first and second computers; and

instructions stored in the memory for intercepting the data packet, determining whether the information stored in the memory of the bridge correlates the first and second computers, and if so, decrypting the data packet to generate a new data packet including a new address header, and transmitting the new data packet onto the second computer.

21. The method of claim 18, wherein the new address header includes information indicating the first computer is a source of the new data packet and the second computer is a destination of the new data packet.

22. (Once Amended) A method for receiving data packets from a first computer to a second computer through a bridge including a processor and a memory that stores instructions for decrypting data packets and information correlating the first and

second computers, the method being carried out according to instructions in the memory of the bridge and comprising:

intercepting a data packet from the first computer to the second computer, the data packet including an address header and a body, the address header including broadcast addresses of the first and second computers and the body including address information representing an internetwork address of the first computer and an internetwork address of the second computer, wherein the address information is encrypted;

determining whether the information stored in the memory of the bridge correlates the first and second computers, and if so, decrypting the data packet to generate a new data packet including a new address header; and

transmitting the new data packet on to the second computer.

23. (Once Amended) The method of claim 22, wherein the body includes the new data packet in encrypted form.

24. (Once Amended) A method for receiving data packets from a first computer to a second computer through a bridge including a processor and a memory that stores instructions for decrypting data packets and information correlating the first and second computers, the method being carried out according to instructions in the memory of the bridge and comprising:

intercepting a data packet from the first computer to the second computer, the data packet including information representing an internetwork address of the first computer and an internetwork address of the second computer;

determining whether the information stored in the memory of the bridge correlates the first and second computers, and if so, decrypting the data packet to generate a new data packet including a new address header; and

transmitting the new data packet on to the second computer;

wherein the data packet includes a header storing key management information providing a mechanism for identifying an encryption method used to encrypt the new data packet.

25. The method of claim 22, wherein the new address header includes information indicating the first computer is a source of the new data packet and the second computer is a destination of the new data packet.

26. (Once Amended) A method of encrypting data packets, comprising:

receiving a data packet from a source for a destination, the data packet including a header section and a data section, the header section storing a source identifier and a destination identifier;

determining whether the data packet should be encrypted upon reference to at least one of the source and destination identifiers;

if the data packet should be encrypted, encrypting the data packet to produce an encrypted data packet; and

generating a new address header and appending the new address header to the encrypted data packet, thereby generating a modified data packet;

wherein the new address header includes a mechanism for identifying an encryption method used to generate the encrypted data packet.

27. (Once Amended) The method of claim 26, further comprising transmitting the modified data packet to the destination.

28. The method of claim 26, wherein the determining whether the data packet should be encrypted comprises accessing stored information that indicates by presence or absence of the source identifier that data packets from the source should be encrypted.

29. The method of claim 26, wherein the determining whether the data packet should be encrypted comprises accessing stored information that indicates by presence or absence of a correlation between the source and destination identifiers that data packets from the source for the destination should be encrypted.

30. (Once Amended) The method of claim 26, wherein the encrypted data packet includes an encrypted data packet header section and an encrypted data packet data section, the encrypted data packet header section including the header section of the data packet after encryption and the encrypted data packet data section including the data section of the data packet after encryption, the modified data packet including a header

portion storing the new address header and a data portion storing the encrypted data packet.

31. The method of claim 30, wherein the encrypted data packet header section stores the source and destination identifiers.

32. (Once Amended) **A method of encrypting data packets, comprising:**

receiving a data packet from a source for a destination, the data packet including a header section and a data section, the header section storing a source identifier and a destination identifier;

determining whether the data packet should be encrypted upon reference to at least one of the source and destination identifiers;

if the data packet should be encrypted, encrypting the data packet to produce an encrypted data packet; and

generating a new address header and appending the new address header to the encrypted data packet, thereby generating a modified data packet;

wherein the encrypted data packet includes an encrypted data packet header section and an encrypted data packet data section, the encrypted data packet header section including the header section of the data packet after encryption and the encrypted data packet data section including the data section of the data packet after encryption, the modified data packet including a header portion storing the new address header and a data portion storing the encrypted data packet;

wherein the source is a host computer in a network and the header portion of the modified data packet stores an identifier of the network.

33. (Once Amended) **A method of encrypting data packets, comprising:**

receiving a data packet from a source for a destination, the data packet including a header section and a data section, the header section storing a source identifier and a destination identifier;

determining whether the data packet should be encrypted upon reference to at least one of the source and destination identifiers;

if the data packet should be encrypted, encrypting the data packet to produce an encrypted data packet; and

generating a new address header and appending the new address header to the encrypted data packet, thereby generating a modified data packet;

wherein the encrypted data packet includes an encrypted data packet header section and an encrypted data packet data section, the encrypted data packet header section including the header section of the data packet after encryption and the encrypted data packet data section including the data section of the data packet after encryption, the modified data packet including a header portion storing the new address header and a data portion storing the encrypted data packet;

wherein the destination is a host computer in a network and the header portion of the modified data packet stores an identifier of the network.

32.
34. The method of claim 26, wherein the source is a host computer or a network.

33.
35. The method of claim 26, wherein the destination is a host computer or a network.

34.
36. (Once Amended) A computer program product adapted for encrypting data packets, comprising:

computer code that when executed causes the reception of a data packet from a source for a destination, the data packet including a header section and a data section, and the header section storing a source identifier and a destination identifier;

computer code that when executed causes the determination of whether the data packet should be encrypted upon reference to at least one of the source and destination identifiers;

computer code that when executed, if the data packet should be encrypted, causes the encryption of the data packet to produce an encrypted data packet;

computer code that when executed causes the generation of a new address header and appends the new address header to the encrypted data packet, the new address header including a mechanism for identifying an encryption method used to generate the encrypted data packet, thereby generating a modified data packet; and

a computer readable medium that stores the computer codes.

~~37.~~ 35 **The computer program product of claim ~~36,~~ 34 wherein the computer readable medium is a memory, random-access-memory, read-only-memory, disk drive, or CD-ROM.**

~~38.~~ 36. (Once Amended) **A computer system for encrypting data packets, comprising:**

a processor;

a computer readable medium coupled to the processor and storing a computer program comprising:

computer code that when executed by the processor causes the processor to receive a data packet from a source for a destination, the data packet including a header section and a data section, and the header section storing a source identifier and a destination identifier;

computer code that when executed by the processor causes the processor to determine whether the data packet should be encrypted upon reference to at least one of the source and destination identifiers;

computer code that when executed by the processor causes the processor to encrypt the data packet to produce an encrypted data packet when it is determined that the data packet should be encrypted; and

computer code that when executed by the processor causes the processor to generate a new address header and append the new address header to the encrypted data packet, thereby generating a modified data packet;

wherein the new address header includes a mechanism for identifying an encryption method used to generate the encrypted data packet.

~~39.~~ 37. **The computer program product of claim ~~38,~~ 36 wherein the computer readable medium is a memory, random-access-memory, read-only-memory, disk drive, or CD-ROM.**

40. (Twice Amended) **A method of decrypting data packets, comprising:**

receiving a data packet from a source for a destination, the data packet including a header section and a data section, the header section storing a source identifier identifying

a broadcast address of the source and a destination identifier identifying a broadcast address of the destination;

determining whether the data packet is encrypted upon reference to at least one of the source and destination identifiers; and

if the data packet is encrypted, decrypting the data packet to produce a decrypted data packet wherein the data section of the data packet includes an encrypted header section and an encrypted data section, and after decryption, the decrypted encrypted header section is used as the header for the decrypted data packet.

41. The method of claim 40, further comprising transmitting the decrypted data packet to the destination.

42. The method of claim 40, wherein the determining whether the data packet is encrypted comprises accessing stored information that indicates by presence or absence of the source identifier that data packets from the source are encrypted.

43. The method of claim 40, wherein the determining whether the data packet is encrypted comprises accessing stored information that indicates by presence or absence of a correlation between the source and destination identifiers that data packets from the source for the destination are encrypted.

44. CANCELED

45. The method of claim 44, wherein the encrypted header section stores the source and destination identifiers.

46. The method of claim 44, wherein the source is a network and the encrypted header section stores an identifier of a host computer in the network.

47. The method of claim 44, wherein the destination is a network and the encrypted header section stores an identifier of a host computer in the network.

48. The method of claim 40, wherein the source is a host computer or a network.

49. The method of claim 40, wherein the destination is a host computer or a network.

50. (Twice Amended) A computer program product adapted for decrypting data packets, comprising:

computer code that when executed causes the reception of a data packet from a source for a destination, the data packet including a header section and a data section, and the header section storing a source identifier identifying a broadcast address of the source and a destination identifier identifying a broadcast address of the destination;

computer code that when executed causes the determination of whether the data packet is encrypted upon reference to at least one of the source and destination identifiers;

computer code that when executed and if the data packet is encrypted, causes the decryption of the data packet to produce a decrypted data packet, wherein the data section of the data packet includes an encrypted header section and an encrypted data section, and after decryption, the decrypted encrypted header section is used as the header for the decrypted data packet; and

a computer readable medium that stores the computer codes.

51. The computer program product of claim 50, wherein the computer readable medium is a memory, random-access-memory, read-only-memory, disk drive, or CD-ROM.

52. (Twice Amended) A computer system for decrypting data packets, comprising:

a processor;

a computer readable medium coupled to the processor and storing a computer program comprising:

computer code that when executed on the processor causes the processor to receive a data packet from a source for a destination, the data packet including a header section and a data section, the header section storing a source identifier identifying a broadcast address of the source and a destination identifier identifying a broadcast address of the destination;

computer code that when executed on the processor causes the processor to determine whether the data packet is encrypted upon reference to at least one of the source and destination identifiers; and

computer code that when executed on the processor causes the processor to if the data packet is encrypted, decrypt the data packet to produce a decrypted data packet, wherein the data section of the data packet includes an encrypted header section and an encrypted data section, and after decryption, the decrypted encrypted header section is used as the header for the decrypted data packet.

53. The computer program product of claim 52, wherein the computer readable medium is a memory, random access memory, read only memory, disk drive, or CD ROM.

38.

54. A system for automatically encrypting and decrypting data packets transmitted from a first host computer on a first computer network, the first host computer having a first processor and a first memory, via an internetwork to a second host computer on a second computer network, the second host computer having a second processor and a second memory, the system including:

security data stored in said first and second memories indicating that data packets meeting at least one predetermined criterion are to be encrypted;

instructions stored in said first memory for determining whether to encrypt one or more data packets, by determining whether said at least one predetermined criterion is met by said one or more data packets;

instructions stored in said first memory for executing encryption of at least a first one of said one or more data packets according to a predetermined encryption/decryption mechanism, when said at least one predetermined criterion is met, for generating a new address header for said first data packet and for appending an encapsulation header to said first data packet and transmitting said first data packet to said second host, said encapsulation header including said new address header and a mechanism for identifying said predetermined encryption/decryption mechanism;

instructions stored in said second memory for receiving said first data packet, determining whether it has been encrypted by reference to said security data in said second memory, and if so then determining which encryption/decryption mechanism was used for encryption, and decrypting said first

data packet by use of said encryption/decryption mechanism.

39.
55. The system as recited in claim 54, 38 wherein said predetermined encryption/decryption mechanism is provided in encrypted form within said encapsulation header.

40.
56. The system of claim 15, wherein said correlation data includes:

encryption rules identifying source and destination networks to and from which packets are to be encrypted; and

host information indicating exceptions to the encryption rules.

41.
57. A system for automatically encrypting data packets for transmission from a first host computer on a first computer network to a second host computer on a second computer network, said first host computer including a first processor and a first memory including instructions for transmitting said data packets from said first host to said second host, the system including:

a bridge computer coupled to the first computer network for intercepting at least a first data packet transmitted from said first computer network, said bridge computer including a second processor and a second memory storing instructions for executing encryption of said first data packet according to a predetermined encryption/decryption mechanism;

information stored in said second memory correlating at least one of the first host computer and the first network with one of the second host computer and the second network, respectively; and

instructions stored in said second memory for intercepting said first data packet before departure from said first network, determining whether said correlation is present, and if so, then executing encryption of said first data packet according to said predetermined encryption/decryption mechanism, generating a new address header including the internetwork broadcast addresses of the first and second computer networks and appending said new address header to said first data packet, thereby generating a modified first data packet on to the second host computer.

42.
58. A computer program product adapted for encrypting data packets, comprising:

computer code that when executed on a computer causes the computer to receive a data packet from a source for a destination, the data packet including a header section and a data section, and the header section storing a source identifier and a destination identifier;

computer code that when executed on a computer causes the computer to determine whether the data packet should be encrypted upon reference to at least one of the source and destination identifiers;

computer code that when executed on a computer causes the computer to, if the data packet should be encrypted, encrypt the data packet to produce an encrypted data packet;

computer code that when executed on a computer causes the computer to generate a new address header storing at least one of a broadcast address associated with the source and a broadcast address associated with the destination, and append the new address header to the encrypted data packet, thereby generating a modified data packet; and

a computer readable medium that stores the computer codes.

43.

~~59.~~ A computer system for encrypting data packets, comprising:

a processor;

a computer readable medium coupled to the processor storing a computer program comprising:

computer code that when executed by the processor causes the processor to receive a data packet from a source for a destination, the data packet including a header section and a data section, the header section storing a source identifier and a destination identifier;

computer code that when executed by the processor causes the processor to determine whether the data packet should be encrypted upon reference to at least one of the source and destination identifiers;

computer code that when executed by the processor causes the processor to if the data packet should be encrypted, encrypt the data packet to produce an encrypted data packet; and

computer code that when executed by the processor causes the processor to generate a new address header storing at least one of a broadcast address associated the source and a broadcast address associated with the destination, and append the

new address header to the encrypted data packet, thereby generating a modified data packet.

60. (Twice Amended) A method of decrypting data packets, comprising:

receiving a data packet from a source at a destination, the data packet including a header section and a data section, the header section storing a source identifier, a destination identifier, and encryption information providing a mechanism for identifying an encryption method used to generate the data packet; and

decrypting the data packet to produce a decrypted data packet, wherein the data section of the data packet includes an encrypted header section and an encrypted data section, and after decryption, the decrypted encrypted header section is used as the header for the decrypted data packet.

61. The method as recited in claim 60, further comprising:

determining from the header section whether the data packet is encrypted; and

wherein decrypting the data packet to produce a decrypted data packet is performed if it is determined that the data packet is encrypted.

62. The method as recited in claim 60, wherein decrypting the data packet to produce a decrypted data packet comprises:

decrypting at least one of the data section of the data packet and the encryption information.

63. CANCELLED

64. (Twice Amended) A computer program product adapted for decrypting data packets, comprising:

computer code that when executed on a computer causes the computer to receive a data packet from a source at a destination, the data packet including a header section and a data section, the header section storing a source identifier, a destination identifier and

encryption information including a mechanism for identifying an encryption method used to generate the data packet;

computer code that when executed on a computer causes the computer to decrypt the data packet to produce a decrypted data packet, wherein the data section of the data packet includes an encrypted header section and an encrypted data section, and after decryption, the decrypted encrypted header section is used as the header for the decrypted data packet; and

a computer readable medium that stores the computer codes.

65. The computer program product as recited in claim 64, further comprising:

computer code that when executed on a computer causes the computer to determine from the header section whether the data packet is encrypted; and

computer code that when executed on a computer causes the computer to decrypt the data packet if it is determined that the data packet is encrypted.

66. The computer program product as recited in claim 64, further comprising:

computer code that when executed on a computer causes the computer to decrypt the data packet using the encryption method.

67. (Twice Amended) A computer system for decrypting data packets, comprising:

a processor;

a computer readable medium coupled to the processor storing a computer program comprising:

computer code that when executed on the processor causes the processor to receive a data packet from a source at a destination, the data packet including a header section and a data section, the header section storing a source identifier, a destination identifier and encryption information including a mechanism for identifying an encryption method used to generate the data packet;

computer code that when executed on the processor causes the processor to determine from the header section whether the data packet is encrypted; and

computer code that when executed on the processor causes the processor to if the data packet is encrypted, decrypt the data packet to produce a decrypted data packet, wherein the data section of the data packet includes an encrypted header

section and an encrypted data section, and after decryption, the decrypted encrypted header section is used as the header for the decrypted data packet.

:

68. The computer system as recited in claim 67, further comprising:

computer code that when executed on a computer causes the computer to decrypt the data packet using the encryption method.

44,

~~69.~~ The system as recited in claim 16, wherein the mechanism indirectly references said predetermined encryption/decryption mechanism.

E'

45.

~~70.~~ The system as recited in claim 20, wherein the mechanism indirectly identifies the encryption method.

46.

~~71.~~ The method as recited in claim 26, wherein the mechanism indirectly identifies the encryption method.

47,

~~72.~~ The computer program product as recited in claim 36, wherein the mechanism indirectly identifies the encryption method.

N.E.

~~73.~~ The computer system as recited in claim 38, wherein the mechanism indirectly identifies the encryption method.
